Contents

Preface ix

Introduction xiii

Part 1. The Basics of SMALLTALK Programming 1

Chapter 1. A Brief Tour of SQUEAK 3

1.1. The first contact 6
1.2. Important points 13
1.3. Exercises 14

Chapter 2. The First Program 17

2.1. Defining new classes 17
2.2. Defining new methods 22
2.3. Program test 31
2.4. Adding methods to subclasses of the `Animal` class 36
2.5. Modification of `Parrot` behavior 39
2.6. Exercises 45
2.7. `self` and `super` 46
2.8. Exercises 54

Chapter 3. Architecture and Inheritance 59

3.1. SMALLTALK software architecture 59
3.2. Static and dynamic inheritance 62

Chapter 4. Some Elements of SQUEAK Syntax and Grammar 67

4.1. Pseudo-variables 67
4.1.1. The pseudo-variable `nil` 67

Copyrighted material

4.1.2. Pseudo-variables `true` and `false` . . . 68
4.1.3. Pseudo-variables `self` and `super` . . . 72
4.1.4. Pseudo-variable `thisContext` . . . 72
4.2. Comments and identifiers . . . 73
4.2.1. Comments . . . 73
4.2.2. Identifiers . . . 74
4.3. Literals . . . 75
4.3.1. Numbers . . . 75
4.3.2. Characters . . . 79
4.3.3. Strings . . . 81
4.3.4. Symbols . . . 86
4.3.5. Arrays . . . 88
4.3.6. Exercises . . . 101
4.4. Variables . . . 105
4.4.1. Temporary variables . . . 107
4.4.2. Method parameters . . . 108
4.4.3. Block parameters . . . 109
4.4.4. Named instance variables . . . 112
4.4.5. Indexed instance variables . . . 114
4.4.6. Global variables . . . 116
4.4.7. Class variables . . . 119
4.4.8. Pool variables . . . 122
4.4.9. Exercises . . . 128
4.5. Messages . . . 131
4.5.1. Unary messages . . . 131
4.5.2. Binary messages . . . 132
4.5.3. Keyword messages . . . 134
4.5.4. Message composition . . . 137
4.5.5. Expression sequences . . . 140
4.5.6. Cascaded messages . . . 141
4.5.7. Primitives . . . 142
4.6. Control structure . . . 146
4.6.1. Conditional control structures . . . 146
4.6.2. Recursive control structures . . . 150
4.6.3. Iterative control structures . . . 155
4.7. Exception handling . . . 175
4.7.1. Exceptions handling messages of the `Object` class . . . 176
4.7.2. Exception handling messages for blocks . . . 183
4.7.3. Messages understood by exceptions . . . 187
4.7.4. Multiple exceptions . . . 188
4.7.5. Define its own exceptions . . . 192
4.8. Exercises . . . 194

Copyrighted material

Part 2. Programming in SQUEAK 201

Chapter 5. Drawings Like in Logo 203

5.1. The Pen class 204
5.2. Some fractals: recursive drawings 212
5.3. Exercises 221
5.4. A restructuring of our Logo programs 222
5.5. A user interface for Logo drawings 226
5.6. Lindenmayer systems 237
5.6.1. A first interpreter of Lindenmayer systems 239
5.6.2. Multiple activations of the same L-system 247
5.6.3. Computing the step size and the initial position 249
5.6.4. Bracketed Lindenmayer systems 257
5.6.5. Read a L-system from a file 259
5.6.6. Exercises 266

Chapter 6. The Dependency Mechanism 271

6.1. Basics of the dependency mechanism 272
6.1.1. Adding and removing dependencies 272
6.1.2. Dependencies management methods 273
6.1.3. A first use case of dependencies 275
6.1.4. A second use case of dependencies 279
6.2. Programming a calculator 291
6.2.1. The keys 292
6.2.2. The calculator 293
6.2.3. Development of a graphical user interface 298
6.2.4. Construction of the interface between views and models 311
6.2.5. An infix calculator 315
6.3. Exercises 321
6.4. A KCalc calculator 322
6.4.1. KCalc description 322
6.4.2. KCalc implementation 323
6.5. Exercises 336
6.6. Concluding remarks 337

Part 3. Appendices 339

Appendix 1 341
A1.1. Exercises from section 1.3 341
A1.2. Exercises from section 2.6 347
A1.3. Exercises for section 2.8 351
A1.4. Exercises from section 4.3.6 359
A1.5. Exercises for section 4.4.9 376

Copyrighted material

Contents

Preface ix

Introduction xiii

Part 1. The Basics of SMALLTALK Programming 1

Chapter 1. A Brief Tour of SQUEAK 3

1.1. The first contact 6
1.2. Important points 13
1.3. Exercises 14

Chapter 2. The First Program 17

2.1. Defining new classes 17
2.2. Defining new methods 22
2.3. Program test 31
2.4. Adding methods to subclasses of the `Animal` class 36
2.5. Modification of `Parrot` behavior 39
2.6. Exercises 45
2.7. `self` and `super` 46
2.8. Exercises 54

Chapter 3. Architecture and Inheritance 59

3.1. SMALLTALK software architecture 59
3.2. Static and dynamic inheritance 62

Chapter 4. Some Elements of SQUEAK Syntax and Grammar 67

4.1. Pseudo-variables 67
4.1.1. The pseudo-variable `nil` 67

Copyrighted material

4.1.2. Pseudo-variables `true` and `false` . . . 68
4.1.3. Pseudo-variables `self` and `super` . . . 72
4.1.4. Pseudo-variable `thisContext` . . . 72
4.2. Comments and identifiers . . . 73
4.2.1. Comments . . . 73
4.2.2. Identifiers . . . 74
4.3. Literals . . . 75
4.3.1. Numbers . . . 75
4.3.2. Characters . . . 79
4.3.3. Strings . . . 81
4.3.4. Symbols . . . 86
4.3.5. Arrays . . . 88
4.3.6. Exercises . . . 101
4.4. Variables . . . 105
4.4.1. Temporary variables . . . 107
4.4.2. Method parameters . . . 108
4.4.3. Block parameters . . . 109
4.4.4. Named instance variables . . . 112
4.4.5. Indexed instance variables . . . 114
4.4.6. Global variables . . . 116
4.4.7. Class variables . . . 119
4.4.8. Pool variables . . . 122
4.4.9. Exercises . . . 128
4.5. Messages . . . 131
4.5.1. Unary messages . . . 131
4.5.2. Binary messages . . . 132
4.5.3. Keyword messages . . . 134
4.5.4. Message composition . . . 137
4.5.5. Expression sequences . . . 140
4.5.6. Cascaded messages . . . 141
4.5.7. Primitives . . . 142
4.6. Control structure . . . 146
4.6.1. Conditional control structures . . . 146
4.6.2. Recursive control structures . . . 150
4.6.3. Iterative control structures . . . 155
4.7. Exception handling . . . 175
4.7.1. Exceptions handling messages of the `Object` class . . . 176
4.7.2. Exception handling messages for blocks . . . 183
4.7.3. Messages understood by exceptions . . . 187
4.7.4. Multiple exceptions . . . 188
4.7.5. Define its own exceptions . . . 192
4.8. Exercises . . . 194

Copyrighted material

Part 2. Programming in SQUEAK 201

Chapter 5. Drawings Like in Logo 203

5.1. The Pen class 204
5.2. Some fractals: recursive drawings 212
5.3. Exercises 221
5.4. A restructuring of our Logo programs 222
5.5. A user interface for Logo drawings 226
5.6. Lindenmayer systems 237
5.6.1. A first interpreter of Lindenmayer systems 239
5.6.2. Multiple activations of the same L-system 247
5.6.3. Computing the step size and the initial position 249
5.6.4. Bracketed Lindenmayer systems 257
5.6.5. Read a L-system from a file 259
5.6.6. Exercises 266

Chapter 6. The Dependency Mechanism 271

6.1. Basics of the dependency mechanism 272
6.1.1. Adding and removing dependencies 272
6.1.2. Dependencies management methods 273
6.1.3. A first use case of dependencies 275
6.1.4. A second use case of dependencies 279
6.2. Programming a calculator 291
6.2.1. The keys 292
6.2.2. The calculator 293
6.2.3. Development of a graphical user interface 298
6.2.4. Construction of the interface between views and models 311
6.2.5. An infix calculator 315
6.3. Exercises 321
6.4. A KCalc calculator 322
6.4.1. KCalc description 322
6.4.2. KCalc implementation 323
6.5. Exercises 336
6.6. Concluding remarks 337

Part 3. Appendices 339

Appendix 1 341
A1.1. Exercises from section 1.3 341
A1.2. Exercises from section 2.6 347
A1.3. Exercises for section 2.8 351
A1.4. Exercises from section 4.3.6 359
A1.5. Exercises for section 4.4.9 376

Copyrighted material